

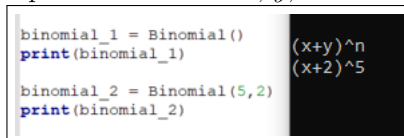
In this lab, we will learn:

1. Counting principles
2. The Binomial Theorem
3. Continued practice of OOP

In this lab, we want to build a class that will create, expand, and evaluate binomials.

A binomial should

- by default be $(x + y)^n$
- allow for changing the values of x , y , or n (even at the time of creation)
- report the values of x , y , or n if asked.

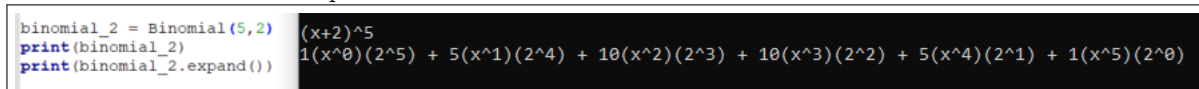


```
binomial_1 = Binomial()
print(binomial_1)

binomial_2 = Binomial(5,2)
print(binomial_2)
```

$(x+y)^n$
 $(x+2)^5$

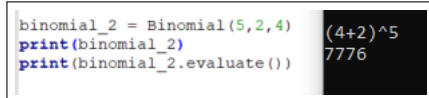
- should have a method to expand a binomial



```
binomial_2 = Binomial(5,2)
print(binomial_2)
print(binomial_2.expand())
```

$(x+2)^5$
 $1(x^0)(2^5) + 5(x^1)(2^4) + 10(x^2)(2^3) + 10(x^3)(2^2) + 5(x^4)(2^1) + 1(x^5)(2^0)$

- should have a method to evaluate a binomial



```
binomial_2 = Binomial(5,2,4)
print(binomial_2)
print(binomial_2.evaluate())
```

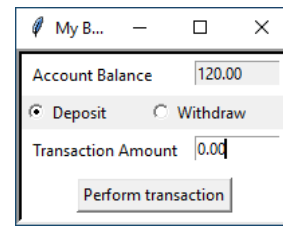
$(4+2)^5$
7776

Once you're done, upload your .py file with your class definitions to D2L.

Challenges: (These are not part of your lab. they're just for fun.)

- Think about the above methods in the context of a general polynomial. Could some of these methods be generalized to a parent class?
- Try implementing a trinomial, deg 4, deg 5, ...

In this lab, we want to create a display for viewing the balance in a bank account. Additionally, a user should be able to make deposits and withdraws, which will update the balance. The following is what my implementation looks like, but you have creative freedom to deviate from this.



The minimal standards you should have are

- A window with
 - The balance (widget and label)
 - The ability to make deposits and withdraws of a specific amount (all of which should be labeled)
 - A button to initiate the transaction
- The balance should only be able to be changed through deposits and withdraws
- The user should not be able to withdraw more than they have. If the user does attempt to do so, they should get a pop-up warning that they have insufficient funds.

To complete this lab you will need a copy of the `breezypythongui.py` module. Please download it from D2L. Documentation for the `breezypythongui.py` module as well the the module itself are provided by Ken Lambert and can be found on his website by clicking [HERE](#). Hint: You will want the documentation!

Alternatively, you can use `tkinter`, which maybe slightly more challenging than the `breezypythonguy`, but it's also the standard. If you choose this option, use <https://docs.python.org/3/library/tk.html> for documentation. Additionally, you may want to [Click Here](#) to read a short article on how to position widgets. The grid method is the one closest to what we did in class.

Once you're done, upload your `.py` file with your class definitions to D2L.

Challenges: (These are not part of your lab. they're just for fun.)

- Multiple bank accounts (think objects and classes)
- Ability to transfer money between accounts
- Ability to enter different types of currency, and have it automatically adjust value to dollars
- A button with counter to pass time and compound interest
- Anything else you think is relevant. Just make sure to leave good comments.